Experiences with Smart City Traffic Pilot

Susanna Pirttikangas, Ekaterina Gilman, Xiang Su, Teemu Leppänen, Anja Keskinarkaus, Mika Rautiainen, Mikko Pyykkönen, Jukka Riekki
Center for Ubiquitous Computing
University of Oulu
Oulu, Finland
e-mail: firstname.lastname@oulu.fi

Abstract— **The infrastructure built in the City of Oulu provides rich information about the city environment and objects moving in it. We utilize this infrastructure in building an IoT system for data-intensive smart city services; by collecting data from real city environment and developing analysis methods for these data. We are building Smart City Traffic Pilot on top of the infrastructure to provide the functionality to collect the data and perform the analysis. Based on this experience, we present in this article requirements for data-intensive smart city services. Moreover, we describe four implemented use cases for utilizing rich data sources available in the smart city: situational picture, driving coach, real time reasoning, and mobile code. A lively collaboration between a large number of different actors is essential in realizing these use cases. Finally, we discuss how the use cases fulfill the requirements and the lessons we have learnt.**

Keywords— IoT, Smart City, digital, data fusion, decision making, Big IoT Data

I. Introduction

Smart cities offer services that are based on rich sets of sensory equipment and data they provide. Internet of Things (IoT) technologies can be used to connect this equipment into a city-scale network of heterogeneous devices that provides the information the services need. However, processing the streaming, near real time, heterogeneous digital data is challenging with ever-increasing data volumes. Moreover, in IoT systems with millions of devices, scalability and interoperability are concerns and centralized solutions cannot operate efficiently anymore.

Within the European Initiative of Smart Cities in 2010-2030[1], transport is one of the three leading topics along with buildings and energy networks. In this paper, we introduce our work on Smart City Traffic Pilot and present four realized use cases. We also list requirements for data-intensive services, discuss how the use cases fulfill these requirements, and present lessons learnt. Our contribution is in studying different approaches for enabling decision making from big, raw, multimodal real time IoT data and providing services based on the fusion of the data. In the first section, we emphasize the importance of pilots in making smart cities a reality. In addition to being a key enabler in our research, our Smart City Traffic Pilot has advanced collaboration among researchers, companies and public organizations and it has helped us to identify the concrete challenges and opportunities of smart city development. Section III gives details on the general requirements we have identified for the data processing pipeline required in Smart City Traffic services and in section IV we present selected use cases that reflect and realize those requirements. In the final section of future work, we emphasize the importance of data models as well as analytics to enable more versatile set of services.

[1]http://setis.ec.europa.eu/set-plan-implementation/technology-roadmaps/european-initiative-smart-cities

II. Building Smart city Traffic Pilot

In the City of Oulu Finland, smart city development was started already over 11 years ago with the set-up of a public wireless LAN network called panOULU, freely open for all citizens and visitors alike. Nowadays, the network has ca. 1300 access points in Oulu and eight nearby towns [1]. The network has subsequently been expanded with a versatile set of sensory equipment, for example, pedestrian mobility can be monitored, among other things, with both Bluetooth (BT) and Wi-Fi data as over 30 000 unique mobile devices use the network each month. This data can be utilized in travel time predictions and when fused to other data, explain different phenomena in the city and distributing this knowledge to people can be done for example through the interactive public displays installed in the city [2].

The city infrastructure is updated frequently as needs to expand or renew are observed (Figure 1). Measurement sites are added to increase the quality, coverage and resolution to scale up the operation of the whole system. For example, the city modernized the traffic control signaling system in 2014. Inductive magnetic loops under the pavement (at most 32 in one intersection, over 123 intersections) are utilized to control the traffic light sequences. The new system can detect and store the traffic light timeframes, in addition to the magnetic loop data, and other digital controlling data from the traffic light poles. Fusing this information to other information gathered from the environment, for example weather, emergency situations, cultural events and social data, enables new services and ways of supporting city traffic engineers [3],[4]. As an example, the traffic center provides a service of green traffic signal pre-emption for ambulances and police vehicles in case of an emergency [5]. In this service, a vehicle location from GPS is sent to the servers every second with the information on the usage of turn signals. This information and advanced

Fig. 1. Installing dual laser measurements sensors and cameras to detect traffic flows, amount of snow, and fleet speed in each lane. A local energy company assists in providing the electricity and connectivity. Companies provide the devices, whereas the data are processed in the companies and research institutes. City of Oulu, 2014.

processing of spatial data is exploited to the on-the-fly controlling of the traffic lights the vehicle will bypass.

As weather data, we utilize the Finnish Meteorological Institute weather information and predictions of detailed current and forthcoming road weather. Most of the data is open but access to the detailed predictions is controlled. Our data sources and the related sampling frequencies are listed in Table I. The Oulu 3D model showing the variety of the information is depicted in Figure 2. All these efforts contribute to Smart City Traffic Pilot – collection of expertise and technological solutions to equip the city with sensing and analyzing capabilities to provide services for citizens.

TABLE I. DATA AVAILABLE IN OULU

Data sources	
Data Source	***Volume/Sampling***
Magnetic loop data	hundreds/5-15 min
Bluetooth access point	tens/real time
Wi-Fi access point	hundreds/real time
Traffic cameras	tens/real time
Parking spaces	tens/real time
Open traffic data	real time
Public transport data (PT)	tens/real time
Road weather data	main roads
Road weather model predictions	2,5 km grids/2 min
Dual laser measurements	tens/real time
Taxi data	hundreds/real time
OBDII	tens/real time
Traffic lights	hundreds/real time
Digital maps	queried when needed
GIS	queried when needed
Other open/crowdsourced data	varies

III. REQUIREMENTS

The transport domain produces in smart cities heterogeneous and voluminous data with different granularity and velocities. For instance, geospatial information about roads and streets as fairly static elements does not change often but the information about moving objects, such as vehicles, changes rapidly. Moreover, some data should be processed on the fly, e.g. data for route recommendations, whereas other services do not require immediate analysis. In many cases, traditional solutions for data storage and analysis are not applicable due to the volume of the data. These challenges require different mechanisms to retrieve, store and process transport-related information. Our Smart City Traffic Pilot is a system of IoT nodes that produce and transmit their data that is refined for further processing and decision making. This section focuses on the requirements that data-intensive services set to such a system.

The obvious requirement is a sufficient selection of data sources (R1). In a real city environment, meeting this requirement requires long-term systematic and collaborative work, both to deploy and maintain the infrastructure providing the data.

To realize the full potential of the data sources, data storage supporting data variety, volume and velocity (R2) is needed in most cases. The storage should support collecting data over a long time span, aggregating different types of structured and unstructured data, and also cleaning and filtering data before storing it. Expert knowledge is required in the cleaning process to make sure all necessary but not redundant information is stored.

Moreover, some data might be available only at the edge devices (or gateways). In a real, operational environment, many partners provide their own servers to ingest and expose data. It might also be necessary to process the data locally as it is not feasible to upload, for example, every second 50 000 samples from a dual laser measurement device to the database. It is application-dependent, what is stored locally, what kind of analysis is done locally and what data features are uploaded to the databases.

Utilizing the data calls for versatile data access (R3) as well. When system components query the data they need (i.e. pull), query interfaces are needed. Moreover, efficient mechanisms are needed for data delivery. The latter alone is sufficient if the data providers send the data when it is available (i.e. push), in this case tools for configuring data delivery are needed instead of query interfaces.

To fulfill the needs of the services, we need rich data analytics (R4). The analysis can be centralized (i.e. in-memory), distributed (i.e. edge analytics), or a combination of these. In all cases, integrating data coming from different sources is necessary in many applications for deriving useful knowledge. The information represented in different formats, various models and with different sampling need to be synchronized. For example, it is a challenge to directly utilize low level data provided by sensors with well-defined knowledge models. Data cleaning plays a central role when data from physical sensors is processed.

For some data, the processing has to include anonymizing and fuzzification to remove sensitive information. This can be done already in the device (e.g. due to person register legislations), or it can be performed after analysis before

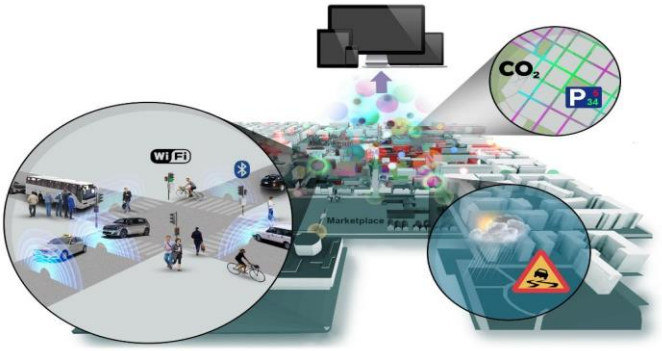

Fig. 2. An illustration of the city-wide sensing in the Smart City Traffic Pilot.

presenting the information derived from the data. The guidelines for privacy are application specific and should be realized by the privacy by design principle [6].

Finally, the system has to support timely decision making. First, decision making calls for comprehensible data visualization (R5). The information produced by the data analysis has to be presented in a clear manner to the users. A promising way of expressing location-dependent data is to utilize virtual worlds. Second, data access and analysis need to have acceptable latency (R6). The cleaned, analyzed and aggregated information needs to be available in time. Moreover, low latency reasoning to refine new knowledge from this information is required. This can lead to distributing reasoning as well, to meet local latency requirements. When data analysis and reasoning is distributed to the edges, performing computations with resource-constrained devices and systems becomes an issue. For instance, it is necessary to consider computing, storage, communication, and energy limitations of IoT devices and protocols and to provide global, scalable, and reliable solutions.

IV. USE CASES

The selected use cases demonstrate end-user services and research enabled by the Smart City Traffic Pilot. The first use case is targeted for a large user group whereas the second one includes more sophisticated data analysis and realizes a recommendation system for individual users. The third use case implements higher level reasoning and the fourth demonstrates the benefits of using mobile code to distribute computation at the edges of the system network.

A. *Situational Picture*

The first use case of Smart City Traffic Pilot offers a situation picture through Web clients [7]. Users can select the type of information they are interested in, for example average fuel consumption calculated on a road segment from the car engine data of several vehicles.

The service is built with the expertise provided by several partners. The equipment and services to track the movement of cars are provided by a telematics company. The information collected from the vehicle engines and location is sent to a database every second through GPRS. The devices are installed into the local taxi company's cabs that operate according to their normal daily routines. We periodically fetch the data from the servers, clean the data, and process it for further analysis.

At this point, we aggregate the data, but a company providing Algorithms-as-a-Service might fulfill this role. The cleaned data is analyzed, geocoded and sent to a map service for visualization. A company that provides integration of transportation related data visualizes the data with other data available. The requirements are fulfilled as follows:

R1. Data sources: moving objects equipped with measurement devices.
R2. Data storage: PostgreSQL database with data cleaning triggers.
R3. Data access: data is pulled from the service providers.
R4. Data fusion: software triggers to calculate statistics from data.
R5. Visualization: MapServer on Apache HTTP server geocodes the data and it can be shown on UIs.
R6. Latency: not yet emphasized.

The visualized statistics include free speed (long time speed average outside rush hours), average speed, average fuel consumption, number of observations, and number of vehicles. In addition, we are developing a solution to visualize the collected information in the virtual 3D model of the city.

B. *Driving* coach

This recommendation system encourages to better driving behavior [8],[9]. The system includes more sophisticated data analysis than the first use case and adds adaptive models that continuously update their operation based on the behavior of the driver. Currently, the analysis is implemented in the R programming language, but to be able to scale the operation to thousands of drivers, we need to address the requirement (R4) with scalable data analysis.

The prototype fuses car engine data, spatial database data, traffic density information from the major local roads, and weather data. Merging all these data together allows extracting essential driving behavior habits, like gear usage patterns, route selections, and aggressive behavior during the drive. The prototype generates general recommendations for fuel-efficient driving, as well as provides some fuel saving calculations.

The information collected is compared to previous trips driven by the user to provide comprehensive statistics. Moreover, a sophisticated rule-based expert system fuses this information with weather information. The system monitors how the user responds to the comments given and adapts itself to provide comments expected to produce more fuel-efficient driving. Fuel economy hints are generated from fuel-consumption models which update themselves to correspond better to changing user context. The system consists of multiple components implemented with diverse technologies as can be seen from the Figure 3.

The requirements are fulfilled as follows:

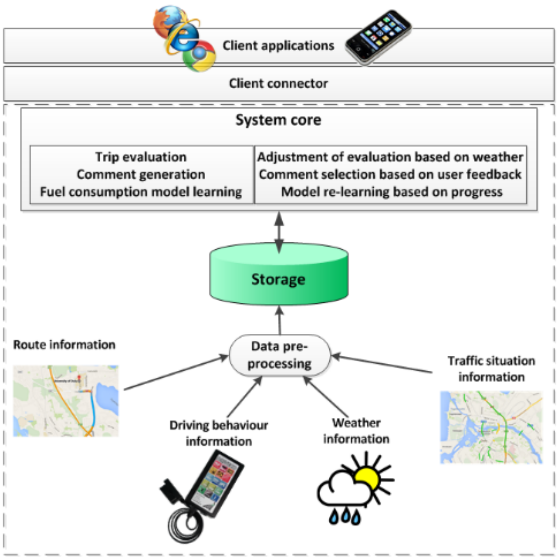

Fig. 3. The driving coach fuses a diverse set of data streaming from multiple service providers. The prototype infers higher level information and creates recommendations expected to lead to more economic driving.

R1. Data sources: engine and location data from a car via GPRS/3G; spatial information about the route driven from a database; weather information from a national service; traffic situation information from a local traffic service.
R2. Data Storage: PostgreSQL database with data cleaning triggers; the storage synchronizes all data.
R3. Data access: data is pulled from service providers.
R4. Data analysis: fuel consumption model learning with R statistical tool; rule-based system generates comments to the user and controls system adaptation.
R5. Visualization: API for application clients, e.g. phone apps and Web services; Web based application delivers information to the user.
R6. Latency: not yet emphasized.

This system demonstrates the potential in fusing diverse data. Visualization can be performed through different services (for example, like in Figure 4 [9]). Driving is a complex and personalized activity, hence understanding what makes it fuel-inefficient and aggressive for each driver has personal as well as societal impact.

C. Reasoning for detecting driving situations

In this use case, a distributed semantic reasoner detects different events and situations from real time data of taxi cabs [10]. This kind of information could be useful for city traffic engineers, citizens and taxi fleet operators, for example. Furthermore, when fused to other data like information about the weather and accident data, predictions can be made.

This prototype delivers IoT data with semantic representations to distributed reasoning engines. An ontology model and a set of rules are designed to reason driving situations. The ontology describes driving activities and traffic conditions, including traffic jams, turns, speeding, stopping for a long time, strong acceleration and deceleration, and areas

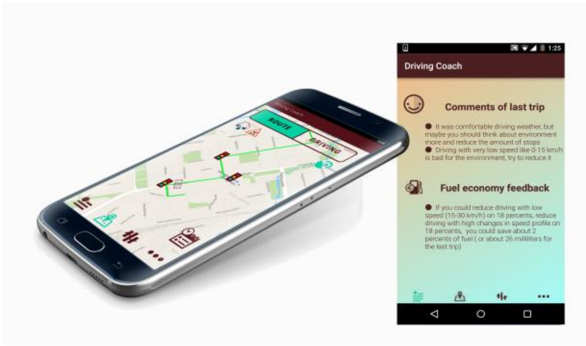

Fig. 4. A mobile user interface showing the results of data analytics, the route of the trip, including both spatial characteristics, as well as driving behavior properties.

where taxis often stop for a while. The ontology is lightweight to minimize the reasoning latency. Table II presents rule examples. For example, the first rule deduces from a velocity value that a car is driving at a low speed. We utilize simple rules to compose complex rules. Moreover, these rules are designed according to the input sequence of individual GPS observations dispatched by the IoT data sources (i.e. cars).

Reasoning engines are implemented with Jena, as it supports a comprehensive subset of OWL language and interprets most of the RDF syntaxes. Information is deduced from consecutive observations by comparing changes in direction and velocity. A sequence of observations is first aggregated and converted to RDF data, then the ontology and rules are applied for decision making.

The focus of this use case is on scalability and efficiency, that is, on minimizing the latency of the whole process of delivering different volumes of data through aggregation and reasoning to storing reasoned facts into the database. The dataset incudes 5 543 348 observations and 72 063 524 RDF triples, from 65000 separate trajectories. An illustration of the reasoning system is depicted in Figure 5.

The requirements are fulfilled as follows:

R1. Data sources: vehicles equipped with measurement devices.
R2. Data storage: RDF database.
R3. Data access: a message broker distributes IoT data with semantic representations to physically distributed reasoning nodes; deduced knowledge can be retrieved from RDF storage.
R4. Data analytics: rule-based reasoning engines process rule sets for detecting driving situations.
R5: Visualization: not yet emphasized.
R6: Latency: message broker balances the load of the distributed reasoning engines.

Our experimental results illustrate the scalability and latency of multiple reasoning nodes. Data formats do not make a big difference with distributed reasoning nodes. The data aggregation strategy has a considerable effect on reasoning performance. When looking at latencies in different stages,

TABLE II. RULE EXAMPLES (SLIGHTLY MODIFIED FROM [9])

Fact	***Triggering rule***
Low speed	Observation hasVelocity<25km/h → LowSpeed
Jam	LowSpeed hasDuration>90s ∧ LowSpeed hasAverageSpeed<20km/h → Jam
Long stop	LowSpeed hasVelocity<3km/h →Stop AND Stop hasDuration>3min → LongStop
High speed	Observation hasVelocity>80km/h → HighSpeed
Speeding	HighSpeed hasVelocity>100km/h → Speeding
Left turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b-90deg ∨ a=b+270deg → LeftTurn
Right turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b+90deg ∨ b=a-270deg → RightTurn
U-turn	LowSpeed[1] hasDirection(a) ∧ LowSpeed[2] hasDirection(b) ∧ a=b-180deg ∨ b=a+180deg → U-Turn
High acceleration	Observation[2] hasVelocity(v2) hasTmeStamp(t2) ∧ (v2-v1)/(t2-t1)>2.5m/s^2 → HighAcc
Crossing zone	LeftTurn hasLocation(x) ∧ RightTurn hasLocation(x) → CrossingZone
Stopping zone	LongStop[1] hasLocation(x) ∧ LongStop[2] hasLocation(x) ∧ LongStop[3] hasLocation(x) → StoppingZone
Jam zone	Jam[1] hasLocation(x) ∧ Jam[2] hasLocation(x) ∧ Jam[3] hasLocation(x) → JamZone
Pollution zone	HighAcc[1] hasLocation(x) ∧ HighAcc[2] hasLocation(x) ∧ HighAcc[3] hasLocation(x) → PollutionZone
Go-slow zone	HighDeacc[1] hasLocation(x) ∧ HighDeacc[2] hasLocation(x) ∧ HighDeacc[3] hasLocation(x) → GoSlowZone

such as data transmission, processing and reasoning, the total latency consists mostly of reasoning and message processing, including aggregation and transmission time and time taken to store results in the RDF database.

D. Mobile code

This prototype focuses on injecting mobile code, as mobile agents, into the Traffic Pilot [11]. The aim is to tackle the issues related to resource-constrained and heterogeneous IoT systems. Mobile agents, as software agents, autonomously control their own execution through migration, act on behalf of system entities, interact with agent and non-agent entities and abstract heterogeneous subsystems. Mobile agent-based task execution improves system robustness, fault tolerance and enables flexibility in system configuration, as communication and computation loads can be distributed into the edges of the network according to the application, task or agent based rules.

Mobile agent-based (one-time or continuous) data queries are facilitated at multiple levels. Service providers and end-users can inject their data queries directly into the system and receive results in real-time. Moreover, data processing at the data source decreases communication costs and latency. The requirements are fulfilled as follows:

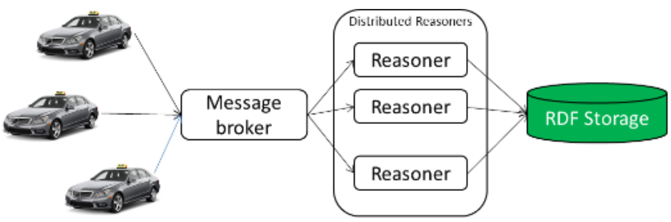

Fig. 5. The database storing the reasoned information has query interface (R3) and support for decision making (R6).

R1. Data sources: the use case does not limit data sources.
R2. Data storage: data is stored at the source, i.e. distributed in the edges of the network.
R3. Data access: queries deployed as cooperative agents in a multi-agent system.
R4. Data analysis: data is analyzed in runtime at the source.
R5. Visualization: mobile agents can provide multiple application or task-specific presentations of the select data features in real-time, encapsulating the raw data into the device.
R6. Latency: computational and communication resources are autonomously utilized where they are available; intermediate results can be locally shared in the system in real-time.

V. DISCUSSION

Traditionally data sources are utilized in silos where each service provider is attached to their own, limited portfolio of data sources and services. These silos need to be demolished to enable Smart Cities. Our Smart City Traffic Pilot is developed and used by the city, different companies and research institutes are involved in this innovative ecosystem. Within the ecosystem, trust is created between partners, they can learn each other's way of doing business and also learn what data, technology and services each partner can offer or could utilize in their business.

From both business and research perspectives, authentic operating environment exposes the concrete challenges and opportunities. Collaboration can be brainstormed in meetings, but it is a completely different story to discuss about collaboration when, for example, two companies have installed their devices in the same environment. For example, our partners found a completely new way of operating traffic cameras when a laser measurement device was installed next to it: the camera is triggered when a vehicle is detected from the laser measurements. This resulted in more accurate results in license plate recognition.

The old technologies within the cities can and should be utilized in the development work, but solutions like data fusion are needed to tackle the limitations. Data transmission and sampling frequencies are not necessarily top-notch, increasing latencies. Some sensors, for example, might be connected to proprietary, closed systems, hence work is needed to provide access to these components. In our Smart City Traffic Pilot, part of the data is produced by almost static, only infrequently updated data sources (e.g. geospatial information). Moreover, some data is produced with low frequency, often by sparsely located sensors. This is the case specifically when old infrastructure is connected to the system. The data storages need to support this large variety of the data streams: in some use case terabytes of data might be stored each second while in another case, most data process might be done at the edges by mobile agents and the storage should have minimal latency. Clearly the Big Data paradigm is not sufficient alone, but edge analytics is required as well.

Exploiting the full functionality of a spatial database often requires using a special platform. Such a platform might have limited support for real time processing. For example, in the smart city scenario, map matching the data – which is in some cases very sparse – in real time and reliably, required algorithms that are not available in the platform we used. A data query against a particular matched route or trajectory can require substantial processing. With our real world experiments, we gather expertise and can participate in developing also these platforms further to match the needs of Smart Cities.

Data models are keys for creating a foundation for a reusable, interoperable IoT in large scale. While IoT protocols are being developed and standardized, data models which enable integration and interoperability of large amounts of information are increasingly important. OMA Lightweight Machine To Machine (OMALWM2M) specifies a set of common interfaces and data models to enable plug and play interoperability between CoAP devices and local or remote services. IPSO Smart Objects introduces extensible application oriented web objects based on OMALWM2M and web object API. World Wide Web Consortium has efforts on developing the capability to exchange rich descriptions of Things and the context information of things. Moreover, large ecosystems are being developed based on protocols and data models. For example, OneM2M is creating a very broad set of standards for interworking with CoAP and OMALWM2M to enable interoperability.

The presented use cases illustrate our ongoing work. Most requirements are fulfilled at an adequate level for the targeted functionality, but more development is required to improve them in Smart City Traffic Pilot. The first use case presented the components required for delivering situational picture of traffic. The work will continue with improving the situational picture as new data sources become available and analysis methods are developed further. In this prototype, we realize the basic preprocessing need for data. An important topic will be to detect automatically, on-the-fly, erroneous measurements and not include them into the databases and further, in the final reasoning from the current situation.

The second use case is aimed for personal use, but the fuel consumption models can use the information from a pre-defined user group similar to the driver, as well. A mobile application was also implemented to test the driver coach recommendation system in logistics to save time and money for a large fleet.

We have not yet targeted the acceptable latency requirement in all the cases of Smart City Traffic Pilot, but only in the reasoning and mobile code use cases. However, this is our current focus as we test our big data platform that supports streaming and batch processing. The ingested data is stored in several data nodes in a distributed file system. The platform already supports periodical data analysis and stores the results into the file system. Analysis is performed with RHadoop-library and Hadoop Map-Reduce programming framework. Currently, we test Spark Streaming capabilities to enable fault-tolerant streaming applications. We identify and implement seven functional phases; ingestion and aggregation; data pre-processing and fusion; real-time analysis; storage and metadata creation; periodic batch analysis; archiving, querying and post analysis and data provisioning. Our current architecture is depicted in Figure 6. More details on the functionalities can be found in [12].

The third use case realizes distributed, low-latency reasoning using off-the-shelf technologies. This kind of higher level knowledge is useful, for example, in the analysis part of driving coach, but also in a variety of applications. The integration of semantic reasoning and big data analysis is a current research topic, as this functionality is not supported in the state of the art, yet.

The fourth use case illustrates the distribution of computing to the edges of the network. For constrained devices it provides an energy-efficient way to disseminate relevant data. For services, it addresses the limitations of bandwidth and connectivity. Integrating edge processing with big data analysis is a promising research direction, as it addresses application-based data processing needs and enables location-based cooperation of different system entities.

VI. CONCLUSIONS

It goes without saying that data analytics is needed in the whole data processing chain from collecting the data to service provision. The data needs to be filtered, cleaned, the essential features need to be extracted – sometimes automatically, and higher level knowledge needs to be derived from the preprocessed data. We have explored different kinds of data analytics in our use cases. The necessary and basic functionality of cleaning, map matching and deriving simple statistics are realized with software triggers in the first use case of situational pictures. More advanced machine learning algorithms and adaptive models were provided in the second use case of driving coach for streaming data.

We constantly develop visualization tools to enable the connection of virtual and physical world. We can already show the moving object traffic on 3D models in the city of Oulu.

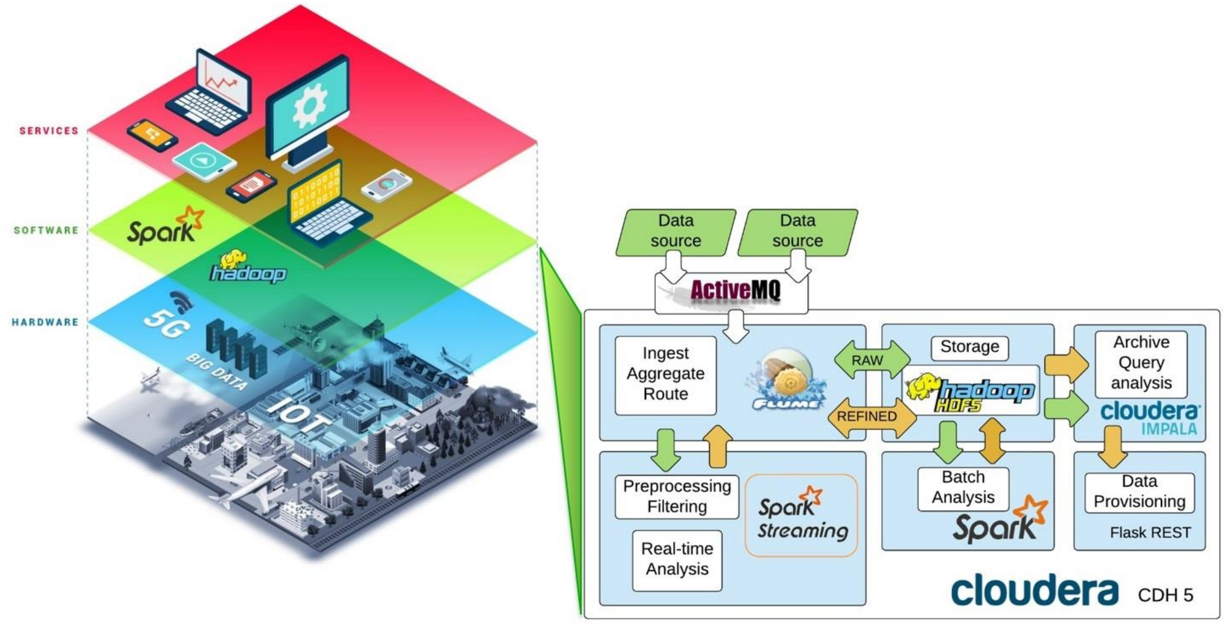

Fig. 6. A Platform for Big Data analytics on a Smart City.

This research also aims at engaging citizens for the development of our home town as a visual effect can be very powerful when showing what is going on in the city. When the listed requirements are fulfilled for a large variety of data analysis tasks, data-intensive services can play a key role in future Smart Cities. We see significant potential in IoT domain in enabling decision making from big real time data.

ACKNOWLEDGMENT

This work was supported by Tekes as part of the Data to Intelligence Program of DIGILE: Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business (currently DIMECC). The companies making all this possible include Infotripla (data integration), Mobisoft (moving object data), Tieto (data analytics platform), Driveco (driving data), Noptel (laser measurement data), FMI Services (weather data), Vidamin (parking data) and Dynniq (traffic signaling data). We would also want to thank the City of Oulu, Oulu Energy, Centria, Finnish Meteorological Institute, Aalto University and University of Tampere.

REFERENCES

[1] T. Ojala, J. Orajärvi, K. Puhakka, I. Heikkinen, J. Heikka, “panOULU: Triple helix driven municipal wireless network providing open and free internet access”, in Proc. 5th International Conference on Communities and Technologies (C&T 2011), Brisbane, Australia, 2011, pp. 118-127.

[2] T. Ojala, V. Kostakos, H. Kukka, T. Heikkinen, T. Lindén, M. Jurmu, S. Hosio, F. Kruger, D.Zanni, “Multipurpose interactive public displays in the wild: Three years later”, in Computer, vol 45, number 5, 2013, pp. 42-49.

[3] V. Kostakos, T. Ojala, T. Juntunen, Traffic in the Smart City: Exploring City-Wide Sensing for Traffic Control Center Augmentation Internet Computing, IEEE, vol 17, number 6, 2013, pp. 22-29.

[4] M. Perttunen, V. Kostakos, J. Riekki, T. Ojala, "Urban traffic analysis through multimodal sensing", in Personal & Ubiquitous Computing. Springer, vol 19, issue 3-4, 2014, pp. 709-721.

[5] J. Talvi, “Always Green at Traffic Lights - for Emergency Vehicles”, in Proc. 20th ITS World Congress, Tokyo, paper No. 2143, 2013.

[6] A. Rohunen, J. Markkula, M. Heikkilä, J. Heikkilä, “Open traffic data for future service innovation – Addressing the privacy challenges of driving data”, in Journal of Theoretical and Applied Electronic Commerce Research, vol 9, number 3, 2014, pp. 71-89. DOI: 10.4067/S0718-18762014000300007

[7] A. Keskinarkaus, E. Gilman, S. Tamminen, M. Hippi, T. Seppänen, J. Riekki, S. Pirttikangas, “Fusion of Road Weather, Moving Object, and Digital Map Data to Support Eco-Driving”, 2016, *submitted*.

[8] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, J. Riekki, “Personalised assistance for fuel-efficient driving”, in Transportation Research Part C: Emerging Technologies, Volume 58, Part D, September, 2015, pp. 681-705, ISSN 0968-090X, http://dx.doi.org/10.1016/j.trc.2015.02.007.

[9] E. Gilman, Y. Zuo, M. Pyykkönen, S. Pirttikangas, J. Riekki, “Delivering eco-driving information to drivers”, in Proc. 11th ITS European Congress, Glasgow, Scotland, 6-9 June 2016.

[10] A.I. Maarala, X. Su, J. Riekki, “Semantic Reasoning for Context-aware Internet of Things Applications”, in IEEE Internet of Things Journal, Vol. 2, Iss. 4, 2016, pp 1-13.

[11] T. Leppänen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Närhi, J. Riekki, T. Ojala, “Mobile Agents for Integration of Internet of Things and Wireless Sensor Networks,” in IEEE International Conference on Systems, Man and Cybernetics (SMC 2013), October 13-16, Manchester, UK, 2013, pp. 14-21.

[12] A.I. Maarala, M. Rautiainen, M. Salmi, S. Pirttikangas, J. Riekki, ”Low Latency Analytics for Streaming Traffic Data with Apache Spark”, in Proc. IEEE International Conference on Big Data, Santa Clara, CA, USA, October 2015, pp. 2855-2858.